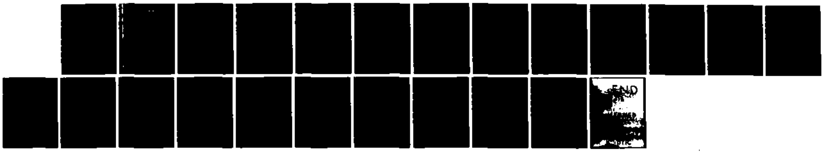
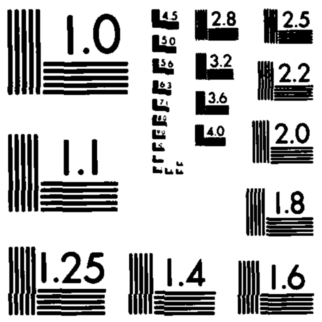
AD-A144 940 SIMULATION USES OF THE EXPONENTIAL DISTRIBUTION(U) 1/1
CALIFORNIA UNIV BERKELEY OPERATIONS RESEARCH CENTER
S M ROSS ET AL. JUN 84 ORC-84-6 AFOSR-TR-84-0819
UNCLASSIFIED AFOSR-81-0122 F/G 12/1 NL





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

12

AFOSR-TR-84-0819

OPERATIONS
RESEARCH
CENTER

# SIMULATION USES OF THE EXPONENTIAL DISTRIBUTION

by

Sheldon M. Ross[†] and Zvi Schechner[††]

ORC 84-6 June 1984

UNIVERSITY OF CALIFORNIA

BERKELEY

# SIMULATION USES OF THE EXPONENTIAL DISTRIBUTION

by

Sheldon M. Ross[†] and Zvi Schechner[††]

ORC 84-6 June 1984

[†]Department of Industrial Engineering and Operations Research, University of California, Berkeley, California.

[††]Department of Industrial Engineering and Operations Research, Columbia University, New York, New York.

This research was supported by the Air Force Office of Scientific Research (AFSC), USAF, under Grant AFOSR-81-0122 with the University of California. Reproduction in whole or in part is permitted for any purpose of the United States Government.

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

| REPORT DOCUMENTATION PAGE | | READ INSTRUCTIONS BEFORE COMPLETING FORM |
|---|---|---|
| 1. REPORT NUMBER<br>ORC 84-6 | 2. GOVT ACCESSION NO.<br>A144 726 | 3. RECIPIENT'S CATALOG NUMBER |
| 4. TITLE *(and Subtitle)*<br>SIMULATION USES OF THE EXPONENTIAL DISTRIBUTION | | 5. TYPE OF REPORT & PERIOD COVERED<br>Technical Report |
| | | 6. PERFORMING ORG. REPORT NUMBER |
| 7. AUTHOR(s)<br>Sheldon M. Ross and Zvi Schechner | | 8. CONTRACT OR GRANT NUMBER(s)<br>AFOSR-81-0122 |
| 9. PERFORMING ORGANIZATION NAME AND ADDRESS<br>Operations Research Center<br>University of California<br>Berkeley, California 94720 | | 10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS<br>2304/A5 |
| 11. CONTROLLING OFFICE NAME AND ADDRESS<br>United States Air Force<br>Air Force Office of Scientific Research<br>Bolling Air Force Base, D.C. 20332 | | 12. REPORT DATE<br>June 1984 |
| | | 13. NUMBER OF PAGES<br>21 |
| 14. MONITORING AGENCY NAME & ADDRESS *(if different from Controlling Office)* | | 15. SECURITY CLASS. *(of this report)*<br>Unclassified |
| | | 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE |

16. DISTRIBUTION STATEMENT *(of this Report)*

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT *(of the abstract entered in Block 20, if different from Report)*

18. SUPPLEMENTARY NOTES

19. KEY WORDS *(Continue on reverse side if necessary and identify by block number)*

| | |
|---|---|
| Simulation | Random Permutation with Weights |
| Normal | 2-Dimensional Poisson Process |
| Von-Neumann Rejection Technique | Nonhomogeneous |
| Order Statistics | |
| Exponential Distribution | |

20. ABSTRACT *(Continue on reverse side if necessary and identify by block number)*

(SEE ABSTRACT)

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-LF-014-6601

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

## ABSTRACT

This paper indicates how exponential random variables can be efficiently used in a variety of simulation problems. One of the problems is the simulation of order statistics from a normal population. ~~We~~ discuss the general problem of simulating order statistics and then consider the normal case. ~~We~~ start by showing how the Von-Neumann rejection approach via the exponential can be modified to become an efficient algorithm for generating a normal and then present a method for generating normal order statistics. ~~We~~ show how to use the exponential to efficiently simulate random permutations with weights. ~~We~~ then consider the problem of simulating a 2-dimensional Poisson process both for a homogeneous and nonhomogeneous Poisson process.

A-1

# SIMULATION USES OF THE EXPONENTIAL DISTRIBUTION

by

Sheldon M. Ross and Zvi Schechner

## 1. SIMULATING ORDER STATISTICS

Consider the problem of simulating the order statistics $X_{(1)} < X_{(2)} < \dots < X_{(n)}$ from a sample of size $n$ from the continuous distribution $F$. If $F$ is invertible, this can be accomplished by simulating $U_{(1)} < \dots < U_{(n)}$ the order statistics from a sample of $n$ uniform (0,1) random variables and then setting $X_{(i)} = F^{-1}(U_{(i)})$. To generate the $U_{(i)}$, we can simulate $n$ uniform (0,1) random variables $U_1, \dots, U_n$ and then order them. (It should be mentioned at this point that the time necessary for sorting the $U_i$'s is not nearly as large as has been indicated in the simulation literature (see, for instance, [5]). The reason being that the sorting need not be done via a general purpose procedure (such as quicksort which requires on the order of $n \log n$ comparisons) but rather can be done with a procedure that utilizes the fact that the values to be sorted are generated from a uniform distribution (see 5.2.1 of [2]). This leads to an algorithm requiring on the order of $n$ comparisons.) We now present another approach for generating $U_{(1)}, \dots, U_{(n)}$.

Let $X_1, \dots, X_{n+1}$ be independent exponential random variables with rate 1 and interpret $X_i$ as the ith interarrival time of a Poisson process. Now set $S_i = \sum_{j=1}^{i} X_j$ and so $S_i$ is the time of the ith event. Now apply the well-known result that given $S_{n+1}$, $S_1, \dots, S_n$ are distributed as the ordered values of a set of $n$ uniform $(0, S_{n+1})$ random variables. Hence, $\frac{S_1}{S_{n+1}}, \dots, \frac{S_n}{S_{n+1}}$ have the joint distribution of $U_{(1)}, \dots, U_{(n)}$. That is, we have the following algorithm:

Step 1: Generate $n+1$ independent exponential random variables $X_1, \ldots, X_{n+1}$.

Step 2: Set $U_{(i)} = \sum_{j=1}^{i} X_j \Big/ \sum_{j=1}^{n+1} X_j$ , $i = 1, \ldots, n$ .

Remarks:

The above procedure was discussed in [4]. The exponential random variables can be generated either by using $X_i = -\log U_i$ , $U_i$ being uniform on (0,1), or by using any of the other known algorithms.

Suppose now, as in [6], that of the $n$ order statistics, $U_{(1)}, \ldots, U_{(n)}$ we only desire $U_{(i)}, U_{(i+1)}, \ldots, U_{(i+k)}$ . To simulate this set of $k+1$ of the $n$ order statistics, start by simulating $X$ , a gamma random variable with parameters $(i,1)$ (either by summing exponentials when $i$ is small or by using one of the algorithms that is more efficient than this when $i$ is large). Interpret $X$ as the time of the ith event of a Poisson process with rate 1. Now, simulate $k$ exponential random variables with rate 1--call them $X_1, \ldots, X_k$--and set $S_{i+j} = X + \sum_{\ell=1}^{j} X_\ell$ , $j = 1, \ldots, k$ . Now, simulate $Y$ , a gamma random variable with parameters $(n+1-i-k, 1)$ , which will represent the time between the $(i+k)$th and the $(n+1)$st event. Then $\frac{X}{S_{i+k}+Y}, \frac{S_{i+1}}{S_{i+k}+Y}, \ldots, \frac{S_{i+k}}{S_{i+k}+Y}$ will have the desired distribution.

That is, we have the following algorithm for simulating $U_{(i)}, \ldots, U_{(i+k)}$ :

Step 1: Generate $k+2$ independent random variables $X$ , $Y$ , $X_1, \ldots, X_k$ with $X$ being gamma with parameters $(i,1)$ , $Y$ being gamma with parameters $(n+1-i-k, 1)$ , and $X_i$ being exponential with rate 1.

Step 2: Set for $j = 0,1, \ldots, k$

$$U_{(i+j)} = \frac{X + \sum_{\ell=1}^{j} X_\ell}{X + \sum_{\ell=1}^{k} X_\ell + Y} .$$

Even easier than simulating uniform order statistics is simulating exponentially distributed ones. Let $Y_1, \ldots, Y_n$ be independent exponential random variables with rate 1 and set

$$Y_{(1)} = \frac{Y_1}{n}$$

$$Y_{(2)} = Y_{(1)} + \frac{Y_2}{n-1}$$

$$Y_{(j)} = Y_{(j-1)} + \frac{Y_j}{n-j+1}$$

$$Y_{(n)} = Y_{(n-1)} + Y_n .$$

It then follows from the lack of memory property of exponentials plus the fact that the minimum of independent exponentials is exponentially distributed with rate equal to the sum of the rates of the exponentials over which we are minimizing that $Y_{(1)}, \ldots, Y_{(n)}$ are distributed as the order statistics of a set of $n$ independent exponentials with rate 1.

Since $F^{-1}(1 - e^{-Y})$ and $F^{-1}(e^{-Y})$ both have distribution $F$ when $Y$ is exponential with rate 1, it follows that we can simulate a set of $n$ order statistics from $F$ by first simulating the exponential order statistics $Y_{(1)}, \ldots, Y_{(n)}$ and then either setting $X_{(i)} = F^{-1}(1 - e^{-Y}(i))$, $i = 1, \ldots, n$ or setting $X_{(i)} = F^{-1}\left(e^{-Y}(n+1-i)\right)$, $i = 1, \ldots, n$.

Hence, if $F^{-1}(1 - e^{-x})$ or $F^{-1}(e^{-x})$ is as easy or almost as easy to calculate as $F^{-1}(x)$, it is more efficient to first simulate exponential rather than uniform order statistics.

## 2. SIMULATING NORMAL RANDOM VARIABLES

Let $Z$ denote a unit normal random variable and set $X = |Z|$. The Von-Neumann rejection method for simulating $X$, via the exponential distribution, leads to the following well-known algorithm:

(a) Generate independent exponentials with rate 1, $Y_1$ and $Y_2$.

(b) Set $X = Y_1$ if $Y_2 \geq (Y_1 - 1)^2/2$. Otherwise, return to (a).

To improve on this note that, by the lack of memory property of the exponential, it follows that if we reject in Step (b)--that is, if $Y_2 < (Y_1 - 1)^2/2$ or, equivalently, if $Y_1 > 1 + \sqrt{2Y_2}$--then $Y_1 - 1 - \sqrt{2Y_2}$ is exponential with rate 1. Hence, we can use this as one of the exponential values in the next iteration. In fact, even if we accept in (b), we can generate an exponential (independent of $X$) by computing $Y_2 - (Y_1 - 1)^2/2$ which will be exponential with rate 1.

Hence, summing up, we have the following algorithm which generates an exponential with rate 1 and an independent unit normal random variable:

<u>Step 1</u>: Generate $Y_1$, an exponential random variable with rate 1.

<u>Step 2</u>: Generate $Y_2$, an exponential with rate 1.

<u>Step 3</u>: If $Y_2 - (Y_1 - 1)^2/2 > 0$, set $Y = Y_2 - (Y_1 - 1)^2/2$ and go to Step 4. Otherwise, reset $Y_1$ to equal $Y_1 - 1 - \sqrt{2Y_2}$ and return to Step 2.

<u>Step 4</u>: Generate a random number $U$ and set

$$Z = \begin{cases} Y_1 & \text{if } U \leq 1/2 \\ -Y_1 & \text{if } U > 1/2 . \end{cases}$$

The random variables $Z$ and $Y$ generated by the above are independent with $Z$ being normal with mean 0 and variance 1 and $Y$ being exponential with rate 1. (If we want the normal random variable to have mean $\mu$ and variance $\sigma^2$, just take $\mu + \sigma z$.)

Remarks:

(1) As is well known, the above requires a geometric distributed number of iterations of Step 2 with mean $\sqrt{2e/\pi} \approx 1.32$ .

(2) The final random number of Step 4 need not be separately simulated but rather can be obtained from the first digit of any random number used earlier. That is, suppose we generate a random number to simulate an exponential, then we can strip off the initial digit of this random number and just use the remaining digits (with the decimal point moved one step to the right) as the random number. If this initial digit is 0,1,2,3 or 4 (or 0 if the computer is generating binary digits), then we take the sign of $Z$ to be positive and take it negative otherwise.

(3) If we are generating a sequence of unit normal random variables, then we can use the exponential obtained in Step 4 as the initial exponential needed in Step 1 for the next normal to be generated. Hence, on the average, we can simulate a unit normal by generating 1.32 exponentials and computing 1.32 squares and .32 square roots.

## 2.1 Simulating Normal Order Statistics

Suppose we want to simulate the order statistics of $n$ independent unit normal random variables. To do so, we will generate $m$ independent exponentials with rate 1--$W_1, W_2, \ldots, W_m$--and use the above algorithm to generate a binomial distributed number of normals. However, so as to eliminate the need to order these resultant normals, we shall first order the exponentials. We thus have the following approach:

<u>Step 1</u>: Generate independent exponential random variables with rate 1, $W_1, \ldots, W_m$ and set

$$W_{(1)} = W_1/m$$

$$W_{(j)} = W_{(j-1)} + \frac{W_j}{m - j + 1} \;, \quad j = 2, \ldots, m \;.$$

<u>Step 2</u>: Set $k = 0$ , $\ell = 1$ .

<u>Step 3</u>: Generate $Y$ , an exponential random variable with rate 1.

<u>Step 4</u>: If $k = m$ stop; otherwise, reset $k$ to equal $k + 1$ .

<u>Step 5</u>: If $Y - \frac{(W_{(k)} - 1)^2}{2} \geq 0$ , set $X_{(\ell)} = W_{(k)}$ , reset $Y$ to equal

$$Y - \frac{(W_{(k)} - 1)^2}{2} \;,$$

reset $\ell$ to equal $\ell + 1$ and go to Step 4. Otherwise, if $Y - \frac{(W_{(k)} - 1)^2}{2} < 0$ , go to Step 3.

The result of the above will be the random variables $X_{(1)}, \ldots, X_{(R)}$ where $R = \ell - 1$ is binomially distributed with parameters $m$ and $p = \sqrt{\pi/2e}$ . Since the above is equivalent to using the rejection method on the $m$ independent and identically distributed (iid) random variables $W_1, \ldots, W_m$ , it follows that the accepted variables are also iid. Hence, given $R$ , $X_{(1)}, \ldots, X_{(R)}$ are distributed as the order statistics of a set of $R$ random variables having the distribution of $|Z|$ where $Z$ is a unit normal.

However, as $R$ need not equal $n$, we are not finished. If $R > n$, then we need randomly choose $R - n$ of the indices $1, \ldots, R$ and then eliminate the variables having these indices. For instance, if $n = 5$ and $R = 7$ and the random choice of 2 of the first 7 integers is 1 and 4, then the resultant 5 order statistics are $\bar{X}_{(i)}$, $i = 1, \ldots, 5$ where $\bar{X}_{(1)} = X_{(2)}$, $\bar{X}_{(2)} = X_{(3)}$, $\bar{X}_{(3)} = X_{(5)}$, $\bar{X}_{(4)} = X_{(6)}$, $\bar{X}_{(5)} = X_{(7)}$. The best way to randomly choose $R - n$ of the indices $1, \ldots, R$ depends on the relative size of $R - n$ in comparison to $R$. If $R - n$ is small compared to $R$, then we can generate random numbers $U$ and eliminate the index $[RU] + 1$ continuing until $R - n$ distinct integers have been removed. If $R - n$ is moderate in comparison to $R$, then there is also a very simple algorithm for the selection (see Section 3.4.2 of [1]).

If $R < n$, then we can reapply the above steps (with a different value for $m$) and then merge the two sets of order statistics. If this results in greater than $n$ order statistics, then we use the above technique to randomly eliminate certain of them. If the combined set is still of size less than $n$, we again reapply the above steps (with again a different value for $m$) until finally we have at least $n$ order statistics.

The problem of choosing the appropriate value of $m$ when $k$ order statistics need be generated remains. As $E[R] = .7576\,m$ and $\sqrt{\text{Var } R} = .4285\sqrt{m}$, we can be almost certain of having $R \geq k$ if we take $m$ to be approximately equal to $\frac{4}{3}k + \sqrt{4k/3}$. However, perhaps a value around $\frac{4}{3}k$ (which would give one roughly a fifty-fifty chance of having $R \geq n$) would work out better. Numerical experience will be necessary to determine the best $m$ when the order statistics of $k$ of the variables are needed.

Once the above is accomplished, we will have the order statistics of the absolute values of $n$ unit normals. To obtain the order statistics of $n$ unit normals, generate $n$ uniforms--one for each order statistics--and break the list of order statistics of the absolute normals into 2 parts--those having a U-value less than 1/2 and those having it greater than 1/2--keeping the same relative ordering within the 2 lists. The reversed order of the second list, with each element in this list given a negative sign, followed by the first list now yields the order statistics of $n$ unit normals.

Remark:

As in Section 2, the final $n$ uniforms needed to determine the signs of the unit normals need not actually be generated but can be "stripped off" from the random numbers used earlier.

## 3. SIMULATING RANDOM PERMUTATIONS WITH WEIGHTS

Consider an urn containing $n$ balls--the ith of which weighs $w_i$, $i = 1, \ldots, n$. The balls are sequentially removed, without replacement, from the urn according to the following scheme. At each withdrawal, a given remaining ball will be removed with probability equal to its weight divided by the sum of the weights of the remaining balls. We are interested in simulating the random permutation $\underline{I} = (I_1, \ldots, I_n)$ where $I_i$ is the index of the ith ball to be removed.

The following algorithm for simulating $I$ is suggested:

<u>Step 1</u>: Simulate independent exponential random variables $\mathcal{E}_1, \ldots, \mathcal{E}_n$ each having rate 1.

<u>Step 2</u>: Set $X_i = \mathcal{E}_i/w_i$, $i = 1, \ldots, n$.

<u>Step 3</u>: Let $I = (I_1, \ldots, I_n)$ where $I_j$ is the index of the jth smallest of $X_1, \ldots, X_n$.

That the above algorithm indeed works follows from the lack of memory property of the exponential distribution in conjunction with the fact that if $X_i$, $i \geq 1$, are independent exponentials with respective rates $w_i$, then

$$P\left\{X_{i_1} = \min\left(X_{i_1}, X_{i_2}, \ldots, X_{i_k}\right)\right\} = \frac{w_{i_1}}{\sum_{j=1}^{k} w_{i_j}} .$$

<u>Remark</u>:

The above can also be used in the problem of [5] which is concerned with generating a weighted random sample of size $k$ of the $n$ balls.

## 4. SIMULATING A CIRCULAR REGION OF A TWO-DIMENSIONAL POISSON PROCESS

A point process consisting of randomly occurring points in the plane is said to be a two-dimensional Poisson process having rate $\lambda$ if

(a) the number of points in any given region of area $A$ is Poisson distributed with mean $\lambda A$ ; and

(b) the number of points in disjoint regions are independent.

For a given fixed point $\underline{0}$ in the plane, we now show how to simulate events occurring according to a two-dimensional Poisson process with rate $\lambda$ in a circular region of radius $r$ centered about $\underline{0}$ . Let $R_i$ , $i \geq 1$ , denote the distance between $\underline{0}$ and its ith nearest Poisson point, and let $C(a)$ denote the circle of radius $a$ centered at $\underline{0}$ . Then

$$P\left\{\pi R_1^2 > b\right\} = P\{\text{no points in } C(\sqrt{b/\pi})\} = e^{-\lambda b} .$$

Also,

$$\begin{aligned} &P\left\{\pi R_2^2 - \pi R_1^2 > b \mid R_1 = r\right\} \\ &= P\left\{R_2 > \sqrt{(b + \pi r^2)/\pi} \mid R_1 = r\right\} \\ &= P\left\{\text{no points in } C\left(\sqrt{(b + \pi r^2)/\pi}\right) - C(r)\right\} = e^{-\lambda b} . \end{aligned}$$

In fact, the same argument can be repeated to obtain

<u>Proposition 1</u>: With $R_0 = 0$ ,

$\pi R_i^2 - \pi R_{i-1}^2$ , $i \geq 1$ , are independent exponentials with rate $\lambda$ .

In other words, the amount of area that need be traversed to encompass a Poisson point is exponential with rate $\lambda$ . Since, by symmetry, the respective angles of the Poisson points are independent and uniformly distributed over $(0,2\pi)$ , we thus have the following algorithm for simulating the Poisson process over a circular region of radius $r$ about $\underline{0}$ :

Step 1: Generate independent exponentials with rate 1, $X_1, X_2, \ldots,$ stopping at

$$N = \min\left\{n : \frac{X_1 + \ldots + X_n}{\lambda\pi} > r^2\right\} .$$

Step 2: If $N = 1$ , stop, there are no points in $C(r)$ . Otherwise, for $i = 1, \ldots, N-1$ , set

$$R_i = \sqrt{(X_1 + \ldots + X_i)/\lambda\pi} .$$

Step 3: Generate independent uniform (0,1) random variables $U_1, \ldots, U_{N-1}$ .

Step 4: Return the $N - 1$ Poisson points in $C(r)$ whose polar coordinates are

$$(R_i, 2\pi U_i) \quad , \quad i = 1, \ldots, N-1 .$$

The above algorithm which requires on average $1 + \lambda\pi r^2$ exponentials and an equal number of uniform random numbers can be compared with the procedure suggested in [3]. This latter procedure simulates points in $C(r)$ by first simulating $N$ , the number of such points and then uses the fact that, given $N$ , the points are uniformly distributed in $C(r)$ . Hence, the procedure of [3] requires the simulation of $N$ , a Poisson random variable with mean $\lambda\pi r^2$ and must then simulate $N$ uniform points on $C(r)$ , by simulating $R$ from the distribution $F_R(a) = a^2/r^2$ and $\theta$ from uniform $(0,2\pi)$ and must then sort these $N$ uniform values in increasing order of $R$. The main advantage of our procedure is that it eliminates the need to sort.

The above algorithm can be thought of as the fanning out of a circle centered at $\underline{0}$ with a radius that expands continuously from 0 to $r$. The successive radii at which Poisson points are encountered is simulated by noting that the additional area necessary to encompass a Poisson point is always, independent of the past, exponential with rate $\lambda$. This technique can be used to simulate the process over noncircular regions. For instance, consider a nonnegative function $g(x)$ and suppose we are interested in simulating the Poisson process in the region between the x-axis and $g$ with $x$ going from 0 to $T$ (see Figure 1).

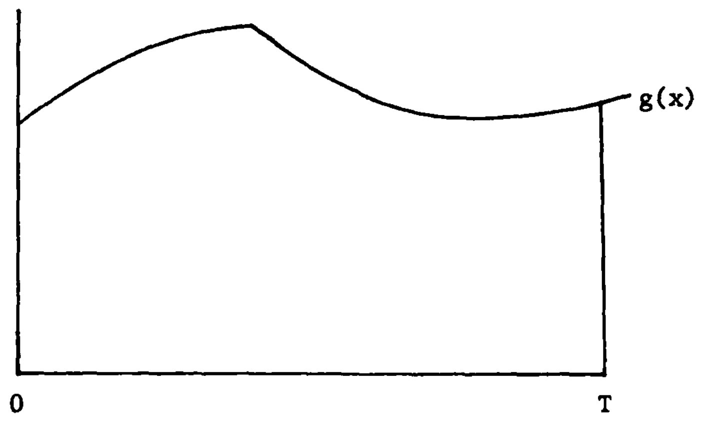


FIGURE 1

To do so, we can start at the left-hand end and fan vertically to the right by considering the successive areas $\int_0^a g(x)dx$. Now if $X_1 < X_2 < \ldots$ denote the successive projections of the Poisson points on the x-axis, then analogous to Proposition 1, it will follow that (with $X_0 = 0$) $\lambda \int_{X_{i-1}}^{X_i} g(x)dx$,

$i \geq 1$ , will be independent exponentials with rate 1. Hence, if we simulate $\mathcal{E}_1, \mathcal{E}_2, \ldots$, independent exponentials with rate 1, stopping at

$$N = \min \left\{ n : \mathcal{E}_1 + \ldots + \mathcal{E}_n > \lambda \int_0^T g(x)dx \right\}$$

and determine $X_1, \ldots, X_{N-1}$ by

$$\lambda \int_0^{X_1} g(x)dx = \mathcal{E}_1$$

$$\lambda \int_{X_1}^{X_2} g(x)dx = \mathcal{E}_2$$

$$\vdots$$

$$\lambda \int_{X_{N-2}}^{X_{N-1}} g(x)dx = \mathcal{E}_{N-1} \ ,$$

and if we now simulate $U_1, \ldots, U_{N-1}$--independent uniform (0,1) random numbers, then as the projection on the y-axis of the Poisson point whose x-coordinate is $X_i$ is uniform on $(0,g(X_i))$ , it follows that the simulated Poisson points in the interval are $(X_i, U_i g(X_i))$ , $i = 1, \ldots, N-1$ .

Of course, the above technique is most useful when $g$ is regular enough so that the above equations can be solved for the $X_i$ . For instance, if

$g(x) = y$ (and so the region of interest is a rectangle), then

$$X_i = \frac{E_1 + \dots + E_i}{\lambda y}, \ i = 1, \dots, N-1$$

and the Poisson points are

$$(X_i, yU_i), \ i = 1, \dots, N-1 .$$

## 5. SIMULATING TWO-DIMENSIONAL NONHOMOGENEOUS POISSON PROCESSES

Consider now a nonhomogeneous Poisson process with intensity function $\lambda(x,y)$ and suppose we are interested in simulating this process over a region $R$. Let $\lambda$ be such that $\lambda(x,y) \leq \lambda$ for all $(x,y) \in R$. A thinning algorithm which first simulates a Poisson process having rate $\lambda$ over R and then accepts the resulting Poisson point $(x,y)$ with probability $\frac{\lambda(x,y)}{\lambda}$ was recommended in [3]. However, we recommend a conditional approach that first simulates $N$, a Poisson random variable with mean $|R| = \iint_{(x,y)\in R} \lambda(x,y)dxdy$--which we assume is calculable (and which represents the number of points in $R$)--and then chooses $N$ points in the region $R$ by simulating from the density $f(x,y) = \frac{\lambda(x,y)}{|R|}$. To simulate from this density, an acceptance rejection procedure that simulates a point $(X,Y)$ uniformly in the region and then accepts it with probability $\lambda(X,Y)/\lambda$ can be used. That is, we have the following algorithm:

Step 1: Simulate $N$, a Poisson random variable with mean $|R|$.

Step 2: Simulate a point $(X,Y)$ uniformly distributed in $R$ and a uniform (0,1) variable $U$ and accept $(X,Y)$ if

$$U \leq \frac{\lambda(X,Y)}{\lambda}.$$

If there have been a total of $N$ points that have been accepted, stop--otherwise, return to Step 2.

Remarks:

(i) Each random point $(X,Y)$ will be accepted with probability $|R|/\lambda A(R)$ where $A(R) = \iint_R dydx$ is the area of $R$. Hence, the mean number of iterations of Step 2 needed to generate $N$ accepted points is $N\lambda A(R)/|R|$ . As $N$ has mean $|R|$ , the above thus needs on average $\lambda A(R)$ iterations of Step 2.

(ii) The method for simulating $(X,Y)$ uniformly in $R$ depends on the geometry of $R$ . One possibility is to enclose $R$ in a rectangle and thus randomly choose a point $(X,Y)$ in this rectangle and then accept it if $(X,Y) \in R$ .

(iii) The fanning out technique of Section 3 can still be employed when $\lambda(x,y)$ is easily integrated and $R$ is "nice." For instance, suppose $R$ is the region of Figure 1. Then if $X_1 < X_2 < \ldots$ denote the successive projections of the Poisson points on the x-axis, then with $X_0 = 0$ , $\int_{X_{i-1}}^{X_i} \int_0^{g(x)} \lambda(x,y)\,dy\,dx$ are independent exponentials with rate 1. Also, given a Poisson point on the line $x = X_i$ , the y-coordinate is distributed according to density

$$\lambda(X_i,y) \Bigg/ \int_0^{g(X_i)} \lambda(X_i,y)\,dy \ .$$

## REFERENCES

1. Knuth, D., Semi-Numerical Algorithms, Volume 2 of The Art of Computer Programming, Second Edition, Addison-Wesley, 1981.
2. Knuth, D., Sorting and Searching, Volume 3 of The Art of Computer Programming, Addison-Wesley, 1972.
3. Lewis, P. A. W. and G. S. Shedler, "Simulation of Nonhomogeneous Poisson Processes by Thinning," Naval Res. Log. Quart., Vol. 26, No. 3, pp. 403-412, 1979.
4. Lurie, D. and H. O. Hurtley, "Machine Generation of Order Statistics for Monte Carlo Simulation," American Statistician, Vol. 26, No. 1, pp. 26-27, 1972.
5. Lurie, D. and R. Mason, "Empirical Investigations of Several Techniques for Computer Generation of Order Statistics," Communications in Statistics, Vol. 2, pp. 363-371, 1973.
6. Ramberg, J. and P. Tadikamalla, "On the Generation of Subsets of Order Statistics," J. Statist. Comput. Simul., Vol. 6, pp. 239-241, 1978.
7. Schucany, W., "Order Statistics in Simulation," J. Statist. Comput. Simul., Vol. 1, pp. 281-286, 1972.
8. Wong, C. K. and M. C. Easton, "An Efficient Method for Weighted Sampling Without Replacement," SIAM J. Comput., Vol. 9, pp. 111-113, 1980.